

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-329093

(43)公開日 平成8年(1996)12月13日

| (51)Int.Cl. ⁶ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
|--------------------------|-------|---------|---------------|---------|
| G 0 6 F 17/30 | | 9194-5L | G 0 6 F 15/40 | 3 1 0 C |
| 9/44 | 5 6 0 | 7737-5B | 9/44 | 5 6 0 B |
| | | 9194-5L | 15/40 | 3 5 0 B |
| | | 9194-5L | 15/411 | 3 1 0 |

審査請求 未請求 請求項の数 9 O L (全 14 頁)

(21)出願番号 特願平7-131793

(22)出願日 平成7年(1995)5月30日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 池ノ谷 和幸

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

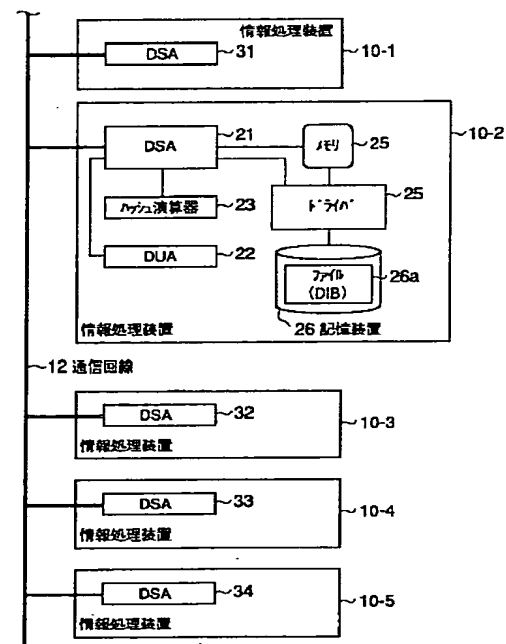
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 分散ディレクトリシステム及び知識情報変更方法

(57)【要約】

【目的】ディレクトリ情報木(DIT)における葉以外のエントリについての相対識別名(RDN)の変更を可能にする。

【構成】情報処理装置10-2は、システムがDIT(ディレクトリ情報木)によるデータ構造によって管理する全情報の少なくとも1部を含む部分DITの情報が格納されたDIBと、DIBから情報を検索するDSA21と、情報の検索要求をDSA21に伝達するDUA22とを有し、DSA21は、何れの情報処理装置におけるDSAがDIT中の何れの部分を管理しているかを示す知識情報を格納するための知識エントリを配置して管理し、部分DIT中の相対識別名を変更する際、知識エントリ中の知識情報に基づいて、他の情報処理装置におけるDSAによって管理される知識情報に影響を与えることが判別された場合に該当する他の情報処理装置のDSAに対して変更内容を含むエントリ更新要求を発行する。



【特許請求の範囲】

【請求項 1】 ITU-T 勧告 X.500 シリーズに準拠し、複数の情報処理装置が通信回線を介して接続された分散ディレクトリシステムにおいて、

前記情報処理装置は、

システムが DIT (ディレクトリ情報木) によるデータ構造によって管理する全情報の少なくとも 1 部を含む部分 DIT の情報が格納された DIB (ディレクトリ情報ベース) と、前記 DIB から情報を検索する DSA (ディレクトリシステムエージェント) と、情報の検索要求を前記 DSA に伝達する DUA (ディレクトリ利用者エージェント) とを有し、

前記 DSA は、

部分 DIT 中に何れの情報処理装置における DSA が前記 DIT 中の何れの部分を管理しているかを示す知識情報を格納するための知識エントリを配置して管理し、自 DSA が管理する部分 DIT のエントリを識別するための相対識別名を変更する際、前記知識エントリ中の知識情報に基づいて、他の情報処理装置における DSA によって管理される知識情報に影響を与えることが判別された場合に、該当する他の情報処理装置における DSA に対して変更内容を含むエントリ更新要求を発行する手段を具備したことを特徴とする分散ディレクトリシステム。

【請求項 2】 前記知識情報として、自己知識、上位知識、及び下位知識をそれぞれ属性として格納し、

前記 DSA は、自 DSA が管理する部分 DIT のエントリの識別名の変更が、前記上位知識または前記下位知識により、DIT 上で隣接する DSA の知識情報に影響を与えるか否かを判別することを特徴とする請求項 1 記載の分散ディレクトリシステム。

【請求項 3】 前記 DSA は、エントリ更新要求を受けた場合に、変更内容に応じて自 DSA が管理する知識情報を変更すると共に、自知識エントリ中の知識情報に基づいて、さらに他の情報処理装置における DSA によって管理される知識情報に影響を与えることが判別された場合に、該当する他の情報処理装置における DSA に対して変更内容を含むエントリ更新要求を発行することを特徴とする請求項 1 記載の分散ディレクトリシステム。

【請求項 4】 前記相対識別名の列による識別名からなる知識情報、及び前記相対識別名の列による識別名によって検索対象を指定する検索要求のそれぞれについて、各相対識別名に対応するハッシュ値の列からなるハッシュ識別名を求めるハッシュ識別名生成手段を具備し、前記 DSA は、

前記検索要求に対して、前記ハッシュ識別名識別手段によって得られたハッシュ識別名を用いて、知識エントリに格納された知識情報から最適な知識情報を検索することを特徴とする請求項 1 記載の分散ディレクトリシステム。

【請求項 5】 前記相対識別名を変更した場合に、相対識別名が変更されたエントリに対して変更前の相対識別名を持つ別名エントリを作成する手段を具備し、

相対識別名を変更した後、新旧 2 つの識別名でのアクセスを可能としたことを特徴とする請求項 1 記載の分散ディレクトリシステム。

【請求項 6】 ITU-T 勧告 X.500 シリーズに準拠し、複数の情報処理装置が通信回線を介して接続された分散ディレクトリシステムにおいて、

前記情報処理装置は、

システムが DIT (ディレクトリ情報木) によるデータ構造によって管理する全情報の少なくとも 1 部を含む部分 DIT の情報が格納された DIB (ディレクトリ情報ベース) と、前記 DIB から情報を検索する DSA (ディレクトリシステムエージェント) と、情報の検索要求を前記 DSA に伝達する DUA (ディレクトリ利用者エージェント) とを有し、

前記 DSA は、

部分 DIT 中に何れの情報処理装置における DSA が前記 DIT 中の何れの部分を管理しているかを示す知識情報を格納するための知識エントリを配置して管理し、自 DSA が管理する部分 DIT のエントリを識別するための相対識別名を変更する際、前記知識エントリ中の知識情報に基づいて、他の情報処理装置における DSA によって管理される知識情報に影響を与えることが判別された場合に、該当する他の情報処理装置における DSA に対して変更内容を含むエントリ更新要求を発行することを特徴とする知識情報変更方法。

【請求項 7】 前記 DSA は、エントリ更新要求を受けた場合に、変更内容に応じて自 DSA が管理する知識情報を変更すると共に、自知識エントリ中の知識情報に基づいて、さらに他の情報処理装置における DSA によって管理される知識情報に影響を与えることが判別された場合に、該当する他の情報処理装置における DSA に対して変更内容を含むエントリ更新要求を発行することを特徴とする請求項 6 記載の知識情報変更方法。

【請求項 8】 前記相対識別名の列による識別名からなる知識情報、及び前記相対識別名の列による識別名によって検索対象を指定する検索要求のそれぞれについて、各相対識別名に対応するハッシュ値の列からなるハッシュ識別名を求め、

前記検索要求に対して、前記ハッシュ識別名を用いて知識エントリに格納された知識情報から最適な知識情報を検索することを特徴とする請求項 6 記載の知識情報変更方法。

【請求項 9】 前記相対識別名を変更した場合に、相対識別名が変更されたエントリに対して変更前の相対識別名を持つ別名エントリを作成することを特徴とする請求項 6 記載の知識情報変更方法。

50 【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ITU-T勧告X.500シリーズ(1988年版)で規定している分散ディレクトリシステム及び知識情報変更方法に関する。

【0002】

【従来の技術】ディレクトリシステムは大規模なネットワーク上に分散されたネットワーク資源を一元管理する機能を提供するサービスである。このディレクトリシステムを用いてネットワーク資源に関する情報を検索したり、更新したりすることができる。

【0003】ディレクトリシステムは、図12に示すように、ネットワーク情報が格納されたDIB(ディレクトリ情報ベース)をアクセスするDSA(ディレクトリシステムエージェント)と、利用者からの要求をDSAに伝達するDUA(ディレクトリ利用者エージェント)から構成されている。

【0004】図12に示すように、ネットワークに分散されたDIBから構成される分散ディレクトリでは、マシンAのDIBにないネットワーク情報は他のマシン(マシンB、もしくはマシンC)に存在する。

【0005】その様な場合、マシンAの利用者はDUAを介して、検索要求をDSAに送信する。DSAは、要求されたネットワーク情報が存在するマシン(この場合は、マシンCとする)に検索を依頼する。検索要求を受けたマシンCは、要求する検索処理を実行し、その結果をマシンAのDSA及びDUAを介して、利用者に返す。

【0006】一方、DIBの論理的データ構造は、図13に示すように、DIT(ディレクトリ情報木)と呼ばれる木構造になっており、その木の節にはエントリが配置され、ネットワーク情報(メールアドレスなど)は属性としてエントリに格納されている。

【0007】各エントリには名前付けがされており、その名前は相対識別名(RDN)で表せる。RDNは、{属性型:属性値}の組み合わせからなり、例えば会社名「会社B」のエントリは、{「組織名」:「会社B」}というRDNを持っている。

【0008】また、DITのルートから任意のエントリまでのRDNの順序列は、識別名(DN)で表せる。例えば、「会社B」のエントリは、({「国名」:「日本」}/{{「組織名」:「会社B」}})のDNを持っている。利用者が「会社B」に所属する従業員の電子メールアドレスを検索しようとする場合、このRDNの列から構成されるDNを指定することによって必要とするエントリを特定することができる。

【0009】分散ディレクトリシステムは、検索要求の際に指定されたDNをもとに、DITのルートから「日本」、「会社B」とたどり、「会社B」の従業員である「池谷和幸」、「橋本浩一」、「増尾洋」のエントリから電子メールアドレスを獲得する。

【0010】また、識別名に別の名前を与える別名エントリを配置することによって、ある1つのエントリを様々な名前で識別することができる。例えば、図13中に示すように、別名エントリ「会社X」を配置することにより、「池谷和幸」のエントリを({「国名」:「日本」}/{{「組織名」:「会社X」}/{{「一般名」:「池谷和幸」}}})という名前でもアクセスすることができる。

【0011】また、分散ディレクトリでは、どのDSAがDIT上のどの情報を管理しているかという知識情報を持たなければならない。知識情報には、上位知識、下位知識、及び自己知識(自分がDITのどの部分を持っているかを示す知識)があり、上位知識を除く、知識はそのDSAが持つ部分木の頂点エントリの識別名とそのDSAのアドレス情報から構成される。上位知識は、上位DSAのアドレス情報のみから構成される。

【0012】これらの知識情報を各DSAが管理することによって、DIBが分散されている分散ディレクトリにおいて、何れのDSAからも目的のエントリを検索することができる。

【0013】一方で、知識情報に影響を与えるDITの葉以外のエントリに対して、そのRDNを変更することが禁止されている。すなわち、葉以外のエントリのRDN変更を許可すると、RDNを変更したエントリの直接上位エントリが別のDSAに存在する場合、そのDSAが保存する知識情報(下位知識)と、RDNを変更したエントリを保持するDSAの知識情報(自己知識)が矛盾してしまうためである。

【0014】しかし、例えば、組織体系を表現したDIBにおいて、組織改正などで葉以外のエントリのRDNを変更する必要性は多々ある。その様な場合では、システム管理者がDSAを利用しないローカルな手段でRDNを変更し、それに伴う知識情報の矛盾を各DIBに対して解消しなければならない。従って、大規模な分散ディレクトリでは、膨大な作業が必要となり、逐次、矛盾を解消していくことは不可能に近い。

【0015】

【発明が解決しようとする課題】このように従来の分散ディレクトリシステムでは、各DSA間の知識情報の一貫性を管理する手段がないため、葉以外のエントリのRDN変更は禁止されていた。このような変更を許可すると、RDNを変更したエントリの直接上位エントリが別のDSAに存在する場合、そのDSAが保存する知識情報(下位知識)と、RDNを変更したエントリを保持するDSAの知識情報(自己知識)が矛盾する。また、DIB内に組織などの木構造を構築する場合、組織改正などに対応するため葉以外のエントリのRDNを変更したいというニーズがあり、それに伴う知識管理が必要である。

【0016】また、ディレクトリサービス要求に対し

て、その識別名を持つエントリがどのDSAに存在するかを特定しなければならない。この処理はその識別名を含む最適な知識情報を検索し、発見する処理である。検索キーは識別名であり、文字列を含む比較的大きなデータである。このデータをそのまま比較すると、比較に要する時間がその大きさに比例して増大する。

【0017】また、位置的に遠隔のDSAから複数のDSAを経由して目的のエントリが存在すると予想されるDSAにディレクトリサービス要求が到達したにも関わらず、目的エントリの識別名が変更されたため、その要求を失敗させることは、ディレクトリシステム全体の応答性能を劣化させることになる。さらに、この問題は知識の一貫性を管理する手段でも問題となる。つまり、DIT上で隣接する2つのDSA内の頂点エントリのRDNを同時に変更した場合、知識変更のためにそれぞれに送信した変更要求は識別名が変更されるため失敗し、知識管理自体が失敗する。

【0018】本発明は前記のような事情を考慮してなされたもので、多大な労力を必要とすることなく、葉以外のエントリに対してもRDN変更を可能にし、それに伴う知識情報の一貫性も保持することができる分散ディレクトリシステムを実現する情報処理装置及び方法を提供することを目的とする。

【0019】また、比較的大きなデータである識別名の比較処理時間を短縮し、全体としての検索処理時間を短縮することを可能にする。また、識別名の不一致による操作要求の失敗を回避することを可能にする。

【0020】

【課題を解決するための手段】本発明は、ITU-T勧告X.500シリーズに準拠し、複数の情報処理装置が通信回線を介して接続された分散ディレクトリシステムにおいて、前記情報処理装置は、システムがDIT（ディレクトリ情報木）によるデータ構造によって管理する全情報の少なくとも1部を含む部分DITの情報が格納されたDIB（ディレクトリ情報ベース）と、前記DIBから情報を検索するDSA（ディレクトリシステムエージェント）と、情報の検索要求を前記DSAに伝達するDUA（ディレクトリ利用者エージェント）とを有し、前記DSAは、部分DIT中に何れの情報処理装置におけるDSAが前記DIT中の何れの部分を管理しているかを示す知識情報を格納するための知識エントリを配置して管理し、自DSAが管理する部分DITのエントリを識別するための相対識別名を変更する際、前記知識エントリ中の知識情報に基づいて、他の情報処理装置におけるDSAによって管理される知識情報に影響を与えることが判別された場合に、該当する他の情報処理装置におけるDSAに対して変更内容を含むエントリ更新要求を発行する手段を具備したことを特徴とする。

【0021】また、前記知識情報として、自己知識、上位知識、及び下位知識をそれぞれ属性として格納し、前

記DSAは、自DSAが管理する部分DITのエントリの識別名の変更が、前記上位知識または前記下位知識により、DIT上で隣接するDSAの知識情報に影響を与えるか否かを判別することを特徴とする。

【0022】また、前記DSAは、エントリ更新要求を受けた場合に、変更内容に応じて自DSAが管理する知識情報を変更すると共に、自知識エントリ中の知識情報に基づいて、さらに他の情報処理装置におけるDSAによって管理される知識情報に影響を与えることが判別された場合に、該当する他の情報処理装置におけるDSAに対して変更内容を含むエントリ更新要求を発行することを特徴とする。

【0023】また、前記相対識別名の列による識別名からなる知識情報、及び前記相対識別名の列による識別名によって検索対象を指定する検索要求のそれぞれについて、各相対識別名に対応するハッシュ値の列からなるハッシュ識別名を求めるハッシュ識別名生成手段を具備し、前記DSAは、前記検索要求に対して、前記ハッシュ識別名識別手段によって得られたハッシュ識別名を用いて、知識エントリに格納された知識情報から最適な知識情報を検索することを特徴とする。

【0024】また、前記相対識別名を変更した場合に、相対識別名が変更されたエントリに対して変更前の相対識別名を持つ別名エントリを作成する手段を具備し、相対識別名を変更した後、新旧2つの識別名でのアクセスを可能としたことを特徴とする。

【0025】

【作用】このような構成によれば、分散ディレクトリに対応したDSAを実現する情報処理装置が、DIT（ディレクトリ情報木）の葉以外の部分のエントリのRDN変更を許可した場合に発生する、各DSAが保持する知識情報の矛盾を、知識情報を属性化し、その属性を知識エントリに格納し、知識エントリに対する変更要求によって解消することにより、1988年版X.500シリーズの範囲で、葉以外のエントリに対してもRDN変更を可能にし、それに伴う知識情報の一貫性も保持することができる。

【0026】また、知識情報に含まれる自己知識、上位知識、及び下位知識を参照することで、DIT上で隣接するDSAの知識情報に影響を与えるか否かが判別できる。また、各情報処理装置のDSAは、他のDSAからの知識情報に対する変更要求を受けた場合には、自知識エントリ中の知識情報を変更すると共に、この変更によって、さらに他の情報処理装置における知識情報に変更が必要となった場合には、該当する情報処理装置のDSAに知識情報の変更要求を発行する。従って、知識情報を変更すべき情報処理装置のDSAに対して、順次、変更要求が伝播して一貫性の維持が図られる。従って、各情報処理装置における知識情報に対して、ローカルな手段を用いて逐次、変更する必要がない。

【0027】また、サービス要求に対して要求されたエントリがどのDSAに存在するかを特定するために知識情報を検索するが、その検索キーとしてハッシュ識別名を用い、その値により検索を実行し、一致するものの中で実際の識別名の比較を行うことで、全体としての検索処理時間を短縮することができる。

【0028】また、RDN変更要求によりRDNを変更した場合、そのエントリに対して時限的に変更前のRDNを持つ別名エントリを作成することで、RDN変更実行後、ある一定期間は新旧2つの識別名でのアクセスを可能とし、識別名誤りによるエラーを回避することができる。

【0029】

【実施例】以下、図面を参照して本発明の一実施例を説明する。図1は本発明に係わる分散ディレクトリシステムの概略構成を示すブロック図である。本実施例の分散ディレクトリシステムは、ITU-T勧告X.500シリーズ(1988年版)に準拠しているものとする。

【0030】図1に示すように、分散ディレクトリシステムは、複数の情報処理装置10-1、10-2、…が、通信回線12を介して接続されている。情報処理装置10-2は、DSA(ディレクトリシステムエージェント)21、DUA(ディレクトリ利用者エージェント)22、ハッシュ演算器23、メモリ24、ドライバ25、及び記憶装置26から構成される。

【0031】DSA21は、ネットワーク情報がエントリとして格納されたDIB(ディレクトリ情報ベース)をアクセスするもので、ドライバ25を用いて記憶装置26から読み出す。また、自情報処理装置10-2内のDIBに、検索要求の対象とするエントリが存在せず、他の情報処理装置内に存在する場合には、通信回線12を介して他の情報処理装置(のDSA)に検索処理を要求し、この検索要求に対する検索結果を受けてDUA22に通知する。

【0032】DUA22は、利用者からの要求を受けてDSA21に伝達する。ハッシュ演算器23は、DSA21が受けた検索要求に対して、要求されたエントリが何れの情報処理装置(DSA)に存在するかを特定するために知識情報を検索する際の検索キーを求める。すなわち、ハッシュ演算器23は、検索要求された操作の識別名を構成するRDNの属性値から検索キーとするハッシュ値を求め、各RDNに対応するハッシュ値の列から構成されるハッシュ識別名を生成する。また、ハッシュ演算器23は、検索の対象となる知識情報中の識別名についても、識別名を構成するRDN属性値からハッシュ値を求めて、知識情報におけるハッシュ識別名を生成する(詳細については後述する)。

【0033】メモリ24は、ドライバ25によって記憶装置26中のファイル26aから読み出されたエントリ、DSA21が受けた検索要求操作の識別名、さらに

はハッシュ演算器23によって求められたハッシュ値に基づくハッシュ識別名等を格納するためのものである。メモリ24は、DSA21によって直接、読み書きされる。

【0034】ドライバ25は、DSA21からの指示に応じて、記憶装置26中にファイル26aとして格納されたDIBから、その一部のエントリを読み出してメモリ24に格納する。

【0035】記憶装置26は、DIBをファイル26aとして格納するためのものである。DIBの論理的データ構造は、DIT(ディレクトリ情報木)と呼ばれる木構造になっており、その木の節にはエントリが配置され、ネットワーク情報(メールアドレスなど)が属性としてエントリに格納されている。記憶装置26に格納されるDIBは、システムがDITによるデータ構造によって管理する全情報の少なくとも1部を含む部分DITの情報が格納される。すなわち、複数の情報処理装置における各記憶装置に格納された部分DITに対応する情報の集合により、全情報が構成される。

【0036】各エントリには名前付けがされており、その名前は{属性型:属性値}の組み合わせからなる相対識別名(RDN)で表せる。また、あるエントリを検索する場合には、ルートから任意のエントリまでのRDNの順序列からなる識別名(DN)によって指定する。

【0037】また、何れの情報処理装置のDSAが、DIT上の何れの部分の情報を持っているかという知識情報が、他のエントリと同様にして知識エントリとして格納されている。知識情報には、上位知識、下位知識、及び自己知識(自装置内にDITのどの部分を持っているかを示す知識)がある(詳細については後述する)。

【0038】なお、前述した情報処理装置10-2の構成は、他の情報処理装置も同様に有し、各情報処理装置10-1、10-3、10-4、10-5が、それぞれDSA31、32、33、34を有しているものとする。各情報処理装置10-1、10-2、…は、知識情報に基づいて、通信回線12を介して他の情報処理装置(DSA)間で検索要求の送付、検索結果の受信を行ないDIBのエントリの情報を得ることができる。

【0039】図2は、各情報処理装置10-2、10-3、…が、それぞれ記憶装置26内に保持しているDIBの部分的な内容(部分DIT)を示している。図2中に示すように、情報処理装置10-2、10-3、10-4、10-5とDSA21、DSA32、DSA33、DSA34とがそれぞれ対応しており、DIBが各DSAに分散されて管理される。

【0040】例えば情報処理装置10-2は、図2に示すように、DIBの一部(部分DIT)を記憶装置26aに格納し、DSA21によって管理している。なお、図2中で用いている記号C、O、OU、CNは、属性型を示し、それぞれ国名、組織名、組織単位名、一般名を

表わしている。

【0041】また、知識情報は、以下のようにしてDITに格納する(図3参照)。知識情報は、知識情報を格納するための知識エントリに格納され、DIT上で各DSAが管理する部分DITの頂点エントリの直接下位に葉エントリとして配置する。また、知識エントリのRDNは、固定的に{CN=knowledge}とする。知識情報は、この知識エントリの属性として格納される。知識エントリの属性は、自己知識属性、上位知識属性、及び下位知識属性から構成される。

【0042】自己知識属性は、自DSAがDIT上のどの部分木を管理しているかを表わし、その属性値を自DSAの頂点エントリの識別名とする。上位知識属性は、上位知識を表わす情報であり、その属性値を上位DSAのアドレス情報、及び上位DSAが保持する頂点エントリの識別名とする。

【0043】下位知識属性は、下位知識を表わす情報であり、その属性値を下位DSAのアドレス情報、及び下位DSAが保持する頂点エントリの識別名とする。図3は図2中に示すDSA1(情報処理装置10-1)が管理する部分DITの知識情報を、前述したように、頂点エントリの直接下位に葉エントリとして配置してディレクトリ情報化した例を示している。また、図3に示すような知識エントリが、ドライバ25に読み出されてメモリ24に格納される場合には、知識情報として図4に示すようにして格納されるものとする(詳細については後述する)。

【0044】なお、上記のエントリ及び属性は規格で定義されていないため、使用する団体、もしくは国で新規に登録しなければならない。次に、本実施例の動作について説明する。

【0045】はじめに、図5に示すフローチャートを用いて、相対識別名(RDN)の変更に伴う知識情報に対する管理処理(知識管理処理)について説明する。知識管理処理では、RDNを変更しようとするエントリのDIT上の位置により、知識情報に矛盾をきたすDSAに対して知識変更要求を発行して、知識情報の矛盾を解消するものである。

【0046】DSAプロセス11は、起動時及び知識情報が変更された場合には、図3に示す知識エントリの属性情報を、ドライバ25を用いて読出し、図4に示すようにしてメモリ24上に格納する。すなわち、知識情報は、知識の種別を表わす知識型と、知識属性の識別名と、存在する位置を示すDSAアドレス情報とを、各知識属性に対応づけて格納している。なお、知識型は、CPが自己知識、SUPRが上位知識、SUBRが下位知識を表わしている。

【0047】例えば、管理者が情報処理装置10-1より、情報処理装置10-2のDSA21が管理するDIT中の“会社B”のエントリのRDNを{“組織名”：

“会社C”}に変更したい場合を例にする。

【0048】この場合、情報処理装置10-1のDSA31は、通信回線12を介して、該当するエントリを持つ情報処理装置10-2(DSA21)にRDN変更要求、すなわち(オブジェクト名：/C：日本/O：会社B，新RDN：O：会社C)を送信する。

【0049】RDN変更要求を受信したDSA21は、RDN変更要求のオブジェクト名と最大一致する識別名を持つ知識情報を検索する。この場合、DSA21は、図4に示す知識情報(識別名)中から、オブジェクト名と一致する知識情報41を検索する。DSA21は、該当する知識情報41により自DSAが管理する情報であることを確認し、ドライバ25を用いて、ファイル15からメモリ24上に“会社B”のエントリを読み出す。
【0050】まず、DSA21は、RDNの変更内容が正しいか否かをチェックする(ステップA1)。すなわち、DSA21は、新RDN(O：会社C)が、同レベルに存在するエントリに使用されているかどうかチェックし、同一RDNをもつエントリが存在しないことを確認する。

【0051】ここで、同一RDNをもつエントリが存在した場合には、情報処理装置10-1のDSA31に、エラー報告を通知する(ステップA8)。RDNの変更内容が正しい場合、DSA21は、メモリ24上に読み出した“会社B”のエントリのRDNを、RDN変更要求に従って“会社C”に変更する(ステップA2)。

【0052】さらに、DSA21は、指定されたエントリが頂上エントリであるか否かを判別する(ステップA3)。ここでは、“会社B”のエントリが、知識情報41により情報処理装置10が保持する頂点エントリであることが判別できる。

【0053】この場合、DSA21は、上位DSAの知識エントリに対して、該当する知識属性を変更するため、変更内容を“置換”とするmodifyEntryを発行する(ステップA4)。すなわち、DSA21は、上位DSA(情報処理装置10-1のDSA32)の知識エントリに対して、変更すべき内容(変更情報)を含むmodifyEntryを発行する。

【0054】modifyEntryは、上位知識を示す知識型SUPRをもつ知識情報42よりオブジェクト名が{/C：日本/CN：knowledge}に設定され、変更情報が下位知識属性を{/C：日本/O：会社C)，DSA2}に置換するように設定される。

【0055】また、DSA21は、指定されたエントリの下位が、他のDSA(情報処理装置)に存在するか否かを判別する(ステップA5)。ここでは、“会社B”のエントリには、知識情報43より、下位がDSA33(情報処理装置10-4)に存在することが判別できる。

【0056】この場合、DSA21は、下位DSAの知

識エントリに対して該当する知識属性を変更するため、変更内容を“置換”とするmodifyEntryをDSA33に発行する(ステップA6)。すなわち、DSA21は、下位DSA(情報処理装置10-4のDSA33)の知識エントリに対しても、変更情報を含むmodifyEntryを発行する。

【0057】modifyEntryは、下位知識を示す知識型SUBRを持つ知識情報43よりオブジェクト名が{／C：日本／O：会社B／OU：開発部／CN：knowledge}に設定され、変更情報が上位知識属性を{(／C：日本／O：会社C)、DSA21}に、自己知識属性を{／C：日本／O：会社C／OU：開発部}に置換するように設定される。

【0058】DSA21は、上位もしくは下位DSAに、modifyEntryを発行した結果を待つ(ステップA7)。DSA32(情報処理装置10-3)、及びDSA33(情報処理装置10-4)から、知識情報の変更が成功したことを示す結果が返された場合、DSA21は、メモリ24上の変更された“会社C”のエントリを、ドライブ25を介してファイル15に書き込み、その結果を情報処理装置10-1のDSA31に返す(ステップA9)。DSA31は、図示せぬDUAを介して要求を行なった管理者に通知する。また、DSA32、及びDSA33から、知識情報の変更が失敗(エラー)したことを示す結果が返された場合、情報処理装置10-1のDSA31に、エラー報告を通知する(ステップA8)。

【0059】一方、DSA32は、知識エントリに対するmodifyEntryを受信した場合、変更情報に従って知識情報を変更する。知識情報の更新処理を実行した後、DSA32(情報処理装置10-3)は、起動時と同様に知識エントリの属性をメモリ上に読み出し、この要求以降新しい知識情報を使用する。

【0060】また、DSA33(情報処理装置10-4)は、知識エントリに対するmodifyEntryを受信した場合(ステップB1)、図6に示すフローチャートに従って動作する。

【0061】DSA33は、メモリ上の知識エントリに対して、modifyEntryに含まれる変更情報に応じて、エントリ更新処理を実行する(ステップB2)。そして、DIT上の下位が他のDSAに存在するか否かを判別する(ステップB3)。

【0062】ここでは、下位DSA(DSA4)が存在するため、知識エントリを更新した後、DSA33は、DSA34(情報処理装置10-5)の知識エントリに対して、該当する知識属性を変更するためにmodifyEntryを発行する(ステップB4)。

【0063】modifyEntryは、前述と同様にして、オブジェクト名が{／C：日本／O：会社B／OU：開発部／OU：コンピュータ／CN：knowle

dge}に設定され、変更情報が上位知識属性を{(／C：日本／O：会社C／OU：開発部)、DSA33}に、自己知識属性を{／C：日本／O：会社C／OU：開発部／OU：コンピュータ}に置換するように設定される。

【0064】DSA33は、下位DSA(DSA34)からの処理結果について応答待ちとなる(ステップB5)。DSA33は、知識情報の変更が成功したことを示す結果が返された場合、メモリ上の変更されたエントリを、ドライブを介してファイルに書き込み、結果を情報処理装置10-2のDSA21に返す(ステップB7)。一方、DSA33は、下位DSA34から、知識情報の変更が失敗(エラー)したことを示す結果が返された場合、エラー報告をDSA21に通知する(ステップB6)。

【0065】このようにして、DITの葉以外のエントリのRDNを変更した場合に、各DSAが保持する知識情報の矛盾を、知識情報を属性化し、その属性を知識エントリに格納し、その知識エントリに対してエントリ変更をすることによって解消することができるので、1988年版X.500シリーズの範囲で、葉以外のエントリに対してもRDN変更を可能にし、それに伴う知識情報の一貫性を管理することができる。

【0066】次に、知識情報を変更するサービス要求に対して、要求されたエントリがどのDSA(情報処理装置)に存在するかを特定するために知識情報を検索する際、検索処理時間を短縮する方法について説明する。

【0067】まず、要求された操作のオブジェクト名と最大一致する識別名を有する知識情報を検索する際、要求された操作の識別名を構成するRDNの属性値をハッシュ演算器23に入力してハッシュ値を求める。

【0068】DSA21は、ハッシュ演算器23から出力されたRDNの各属性値に対応するハッシュ値の列からなるハッシュ識別名を、図7に示すように、要求操作の識別名と対応づけてメモリ24上に作成する。

【0069】例えば、DSA21が識別名{／C：日本／O：会社B／OU：開発部／CN：池谷}の要求操作を受信した場合、ハッシュ識別名{02, 77, 21, 02}が求められる。

【0070】また、知識エントリからメモリ24上に読み込まれた知識情報の識別名に対しても、ハッシュ演算器23を用いてハッシュ識別名を求め、図8に示すように、それぞれ識別名と対応づけてメモリ24に格納する。

【0071】ここで、図7及び図8に示すようにハッシュ識別名を求め、検索キーとしてハッシュ識別名を用いて最適な知識情報を検索する際の動作について、図9に示すフローチャートを参照しながら説明する。

【0072】まず、DSA21は、1つの知識情報について、要求操作のハッシュ識別名が、識別名に対応する

ハッシュ識別名を含むか、もしくは一致するかを判別する(ステップC1)。DSA21は、該当するハッシュ識別名がある場合、該当するハッシュ識別名の一致したハッシュ値の数をカウントする(ステップC2)。

【0073】検索範囲の全ての知識情報に対して、前述した処理を実行していなければ(ステップC4)、次の1つの知識情報を取り出し、同様にして該当するハッシュ識別名の一致したハッシュ値の数を求める。

【0074】図8に示す例では、知識情報81については「2」、知識情報82については「1」、知識情報83については「3」のカウント結果が得られる。DSA21は、各知識情報に対するハッシュ値の数から、最大となった知識情報を求める(ステップC5)。ここで、DSA21は、最大となった知識情報について、実際に本来の識別名を比較して、要求操作の識別名が知識情報の識別名を含むか、もしくは一致するかを判別する(ステップC6)。

【0075】条件を満たす場合には、該当する知識情報を検索結果とするが、条件を満たさない場合には、この知識情報を除いて、次にハッシュ値の数が最大の知識情報について(ステップC7)、前述と同様にして、実際に本来の識別名を比較して判別する。

【0076】図7及び図8に示す例では、読み出し要求操作のハッシュ識別名71{02, 77, 21, 02}に対して、ハッシュ識別名の一致したハッシュ値のカウント数が最大となる知識情報は、知識情報83である(3つのハッシュ値が一致する)。この結果は、実際の識別名を比較した結果とも一致している。従って、DSA21は、要求操作に対する最適な知識情報が、知識情報83であると判明し、検索結果とする。

【0077】また、読み出し要求操作は、知識情報の知識型を見ると、SUBR(下位知識)であることによりDSA33に転送される。このようにして、要求操作の識別名と知識情報のそれぞれについてハッシュ識別名を求め、各ハッシュ識別名に対して検索を行ない、その検索結果を用いて本来の識別名による比較を行なって検索結果を得ている。

【0078】この方法を用いると、識別名の比較はハッシュ識別名による比較と、本来の識別名による比較の2段階になるが全体として検索時間を短縮することができる。すなわち、通常、識別名のデータ長は、文字列等を含んでいるため数十バイトになるが、ハッシュ値を1バイト程度にすることにより、ハッシュ識別名による比較は軽い処理負担となり、さらにハッシュ識別名による比較結果を用いることで本来の識別名による比較を少なくできるので、全体として少ない比較処理時間で検索することができる。これにより最適な知識情報の検索に要する時間が短縮される。

【0079】次に、RDN変更要求に対してRDN変更処理実行後、変更前のRDNをもつ別名エントリを自動

的に作成することで、RDNの変更の影響による識別名誤りによるエラーを回避する方法について説明する。この方法による動作を、図10のフローチャートに示している。なお、図10に示すフローチャートは、図5に示すRDN変更による知識管理処理の動作を説明するフローチャートに、別名にエントリ作成処理(ステップD10)を追加したものであるので、共通する処理(ステップD1~D9)についての詳細な説明を省略する。

【0080】DSA21は、ステップD4、D6において、上位DSA及び下位DSAに対してmodify Entryを発行し、その結果、DSA32及びDSA33から知識エントリ更新要求の結果が返された後(ステップD7)、変更前のRDNを持つ別名エントリ、すなわち識別名{/C:日本/O:会社B}を持つ別名エントリを図11に示すように作成する。

【0081】ただし、下位DSAもしくは上位DSAの知識エントリの更新要求で、その変更内容を“置換”ではなく“追加”にする。これにより、RDN変更処理後、RDN変更処理前に用いられている識別名{/C:日本/O:会社B/OU:営業部/CN:鈴木}をもつ操作要求に対しても、正しく“鈴木”のエントリをアクセスすることができる。

【0082】また、RDN変更処理に伴って追加される別名エントリは永久的なものでなく、ある一定期間が経過すると自動的に削除する。その際、上位DSAもしくは下位DSAの知識エントリに対して、この別名に関する知識情報を削除するエントリ更新要求を発行する。

【0083】このようにして、RDN変更要求によりRDNを変更した場合、そのエントリに対して時限的に変更前のRDNを持つ別名エントリを作成することで、RDN変更実行後、ある一定期間であれば新旧2つの識別名でのアクセスを可能とすることができ、識別名誤りによるエラーの低減を図ることができる。

【0084】

【発明の効果】以上詳述したように本発明によれば、分散ディレクトリに対応したDSAを実現する情報処理装置が葉以外のエントリのRDN変更を可能にした場合に発生する各DSAが保持する知識情報の矛盾を、知識情報を属性化し、その属性を知識エントリに格納し、その知識エントリに対するエントリ変更サービスを利用して解消することにより、1988年版X.500シリーズの範囲で、葉以外のエントリに対してもRDN変更を可能にし、それに伴う知識情報の一貫性も保持することができる。

【0085】また、サービス要求に対して要求されたエントリがどのDSAに存在するかを特定するために知識情報を検索する際、検索キーとしてハッシュ識別名を用い、その値により検索を実行し、一致するものの中で実際の識別名の比較を行うことで、比較的大きなデータである識別名の比較処理時間を短縮し、全体としての検索

処理時間を短縮することができる。

【0086】また、RDN変更サービスによりRDNを変更した場合、そのエントリに対して時限的に変更前のRDNを持つ別名エントリを作成することで、RDN変更実行後、ある一定期間は新旧2つの識別名でのアクセスを可能とし、識別名の不一致による操作要求の失敗を回避することができる。

【図面の簡単な説明】

【図1】本発明の一実施例に係わる分散ディレクトリシステムの概略構成を示すブロック図。

【図2】本実施例における各情報処理装置が保持するディレクトリ情報ベース（DIB）の構成を説明するための図。

【図3】本実施例において知識情報を知識エントリとしてディレクトリ情報化した部分DITの一例を示す図。

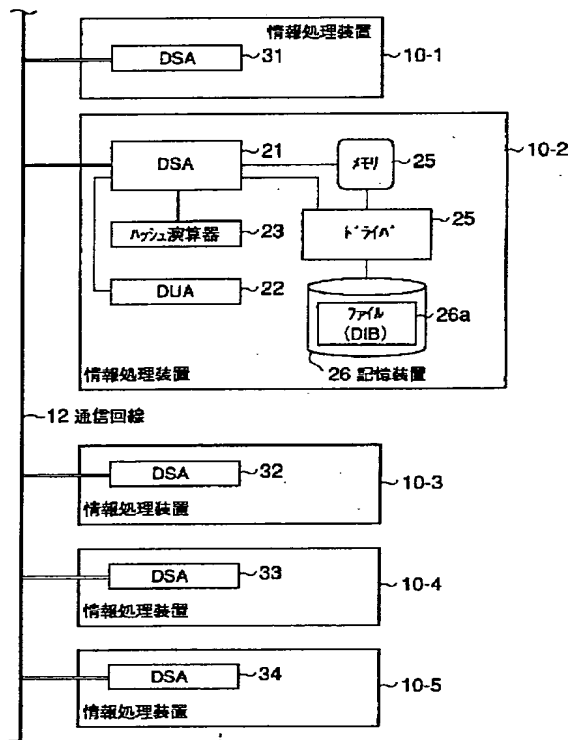
【図4】本実施例においてメモリに格納されている知識エントリから読み出した知識情報の一例を示す図。

【図5】本実施例における相対識別名（RDN）の変更に伴う知識情報に対する管理処理（知識管理処理）の動作を説明するためのフローチャート。

【図6】本実施例における知識変更処理に伴う知識変更処理の動作を説明するためのフローチャート。

【図7】本実施例におけるメモリに格納された知識情報の識別名にハッシュ識別名を追加した知識情報の一例を示す図。

【図1】



* 示す図。

【図8】本実施例におけるメモリに格納された要求操作の識別名にハッシュ識別名を追加した識別名の一例を示す図。

【図9】本実施例におけるハッシュ識別名を用いた知識情報の検索処理の動作を説明するためのフローチャート。

【図10】本実施例における別名エントリの作成処理が附加された相対識別名（RDN）の変更に伴う知識情報に対する管理処理（知識管理処理）の動作を説明するためのフローチャート。

【図11】本実施例における旧RDNをもつ別名エントリが追加されたDIBの一例を示す図。

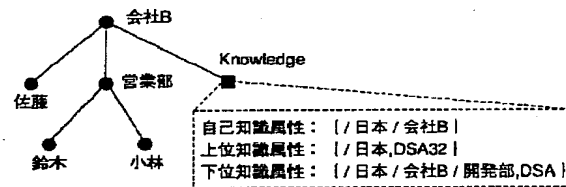
【図12】分散ディレクトリシステムの概略構成を示すブロック図。

【図13】分散ディレクトリシステムにおけるディレクトリ情報ベース（DIB）の論理的データ構造を説明するための図。

【符号の説明】

10-1、10-2、10-3、10-4、10-5…情報処理装置、21…DSA（ディレクトリシステムエージェント）、22…DUA（ディレクトリ利用者エージェント）、23…ハッシュ演算器、24…メモリ、25…ドライバ、26…記憶装置、26a…ファイル。

【図3】

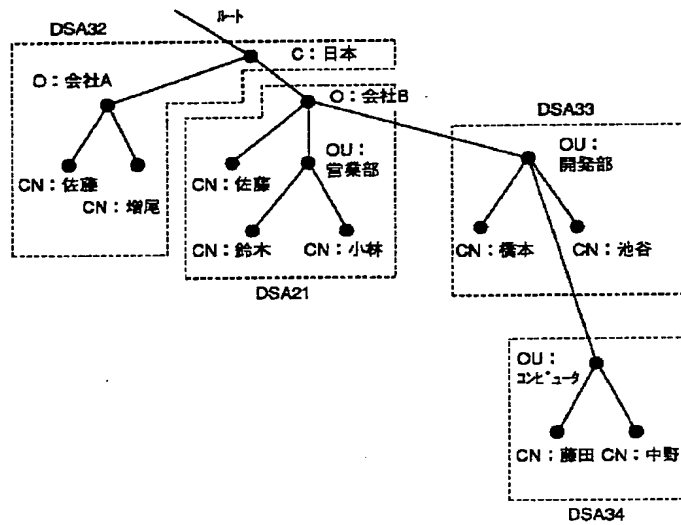


【図4】

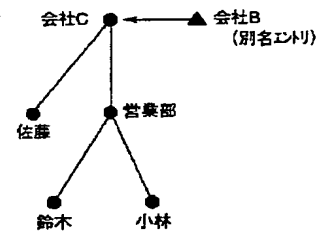
| 知識情報 | | |
|---------|---------------------------|-----------|
| 知識型 | 識別名 | DSA7'12情報 |
| 41 CP | /C: 日本 / O: 会社B | |
| 42 SUPR | /C: 日本 | DSA32 |
| 43 SUBR | /C: 日本 / O: 会社B / OU: 開発部 | DSA33 |

（知識型は知識の種類を表す）
 CP: 自己知識
 SUPR: 上位知識
 SUBR: 下位知識
 CR: 知知知識

【図2】



【図11】



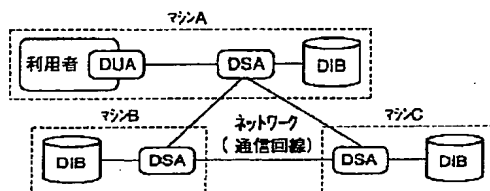
【図7】

| 要求操作の識別名 | |
|------------------|------------------------------------|
| ハッシュ識別名 | 識別名 |
| 71 {02,77,21,02} | /C: 日本 / O: 会社B / OU: 開発部 / CN: 池谷 |

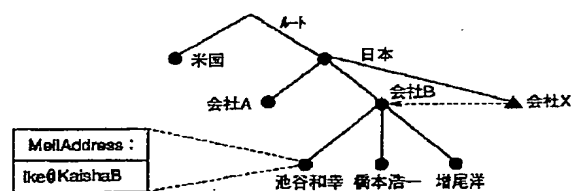
【図8】

| 知識情報 | | | |
|---------|------------|---------------------------|-----------------|
| 知識型 | ハッシュ識別名 | 識別名 | DSA 外に 情報 |
| 81 CP | {02,77} | /C: 日本 / O: 会社B | DSA32 |
| 82 SUPR | {02} | /C: 日本 | |
| 83 SUBR | {02,77,21} | /C: 日本 / O: 会社B / OU: 開発部 | |

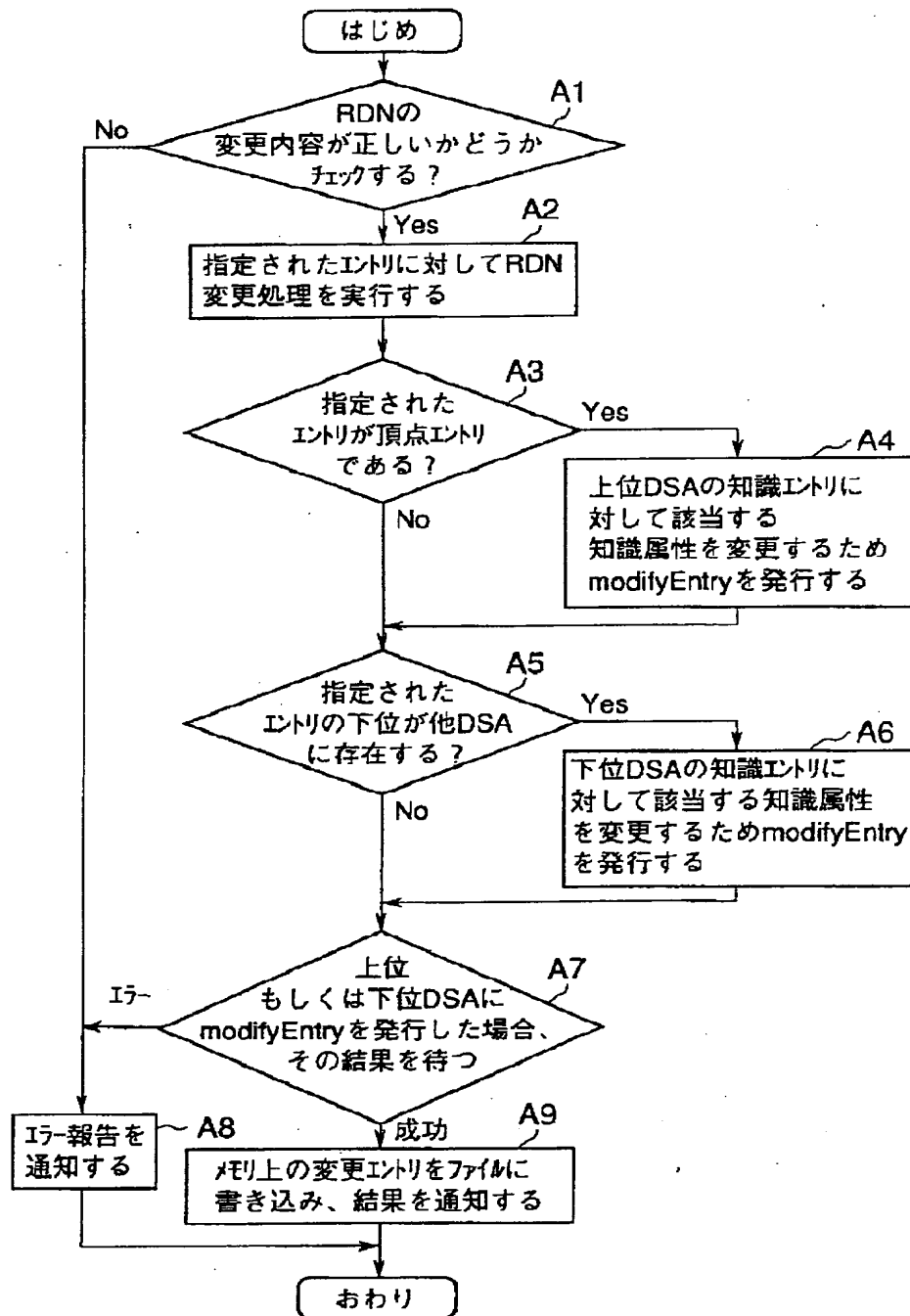
【図12】



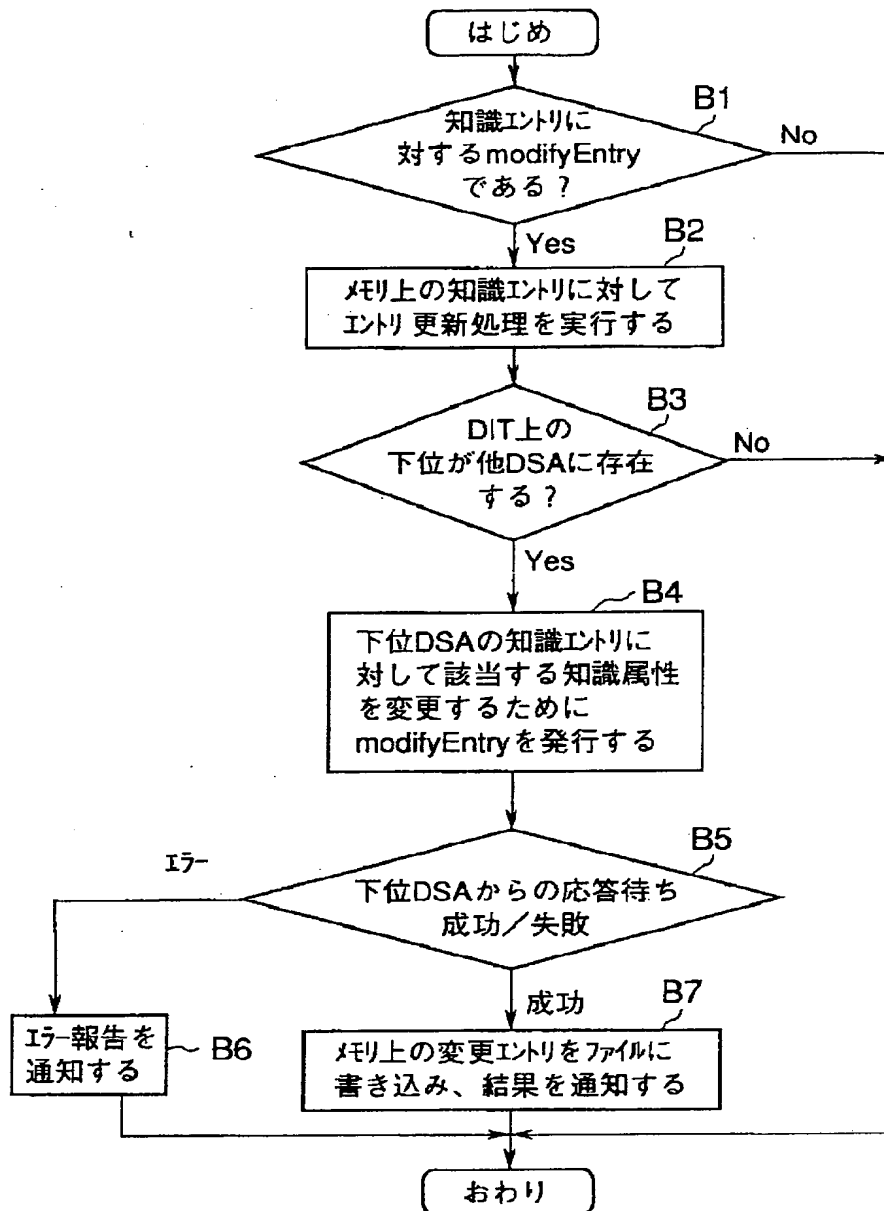
【図13】



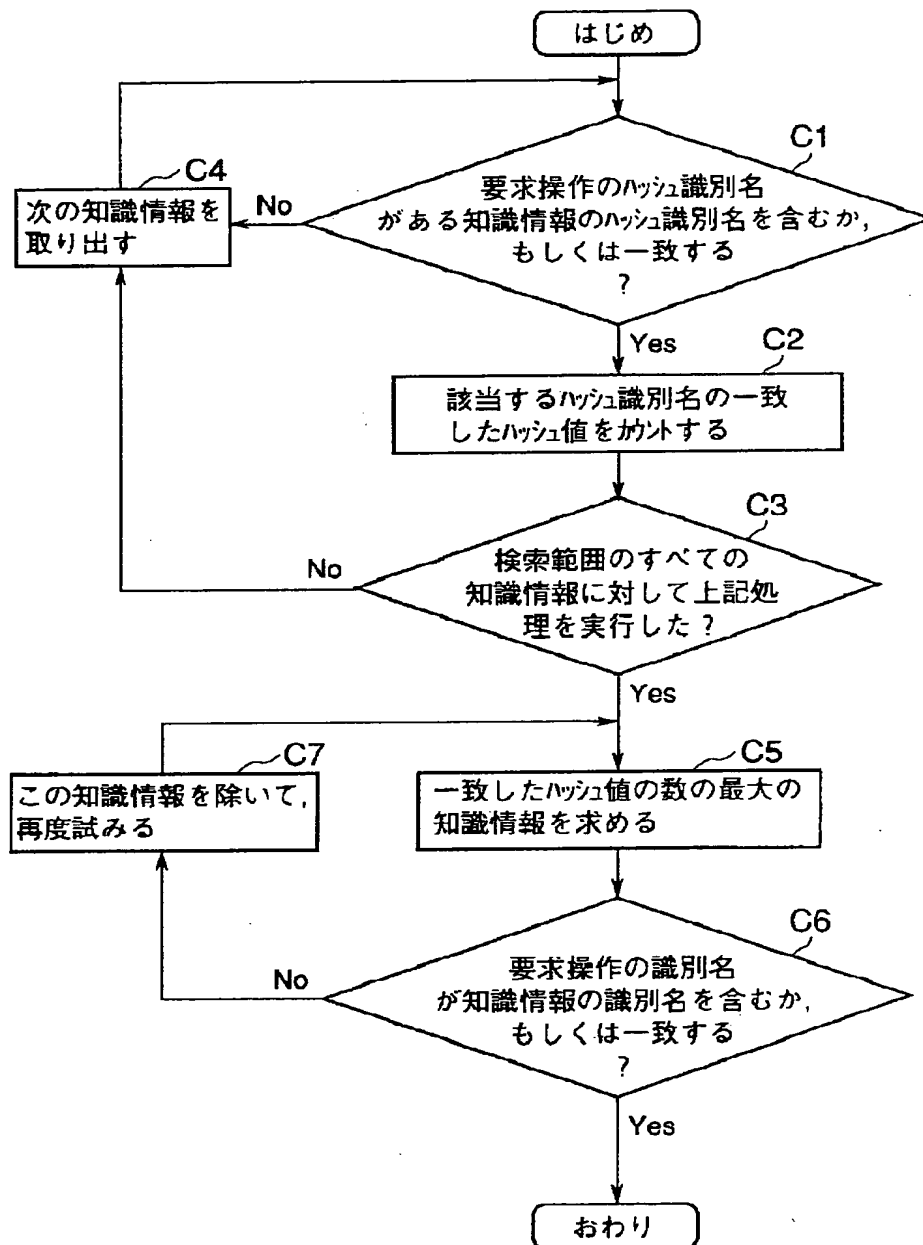
【図5】



【図6】



【図9】



【図10】

